

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-49655
(P2002-49655A)

(43) 公開日 平成14年2月15日 (2002.2.15)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 17/50	6 7 0	G 0 6 F 17/50	6 7 0 K 2 G 1 3 2
G 0 1 R 31/3183		11/22	3 1 0 B 5 B 0 4 6
G 0 6 F 11/22	3 1 0		3 3 0 G 5 B 0 4 8
	3 3 0	G 0 1 R 31/28	Q
11/25		G 0 6 F 11/26	3 1 0
審査請求 未請求 請求項の数10 O L (全 17 頁)			

(21) 出願番号 特願2001-168611(P2001-168611)

(22) 出願日 平成13年6月4日(2001.6.4)

(31) 優先権主張番号 2 0 9 3 4 0

(32) 優先日 平成12年6月5日(2000.6.5)

(33) 優先権主張国 米国 (U S)

(31) 優先権主張番号 8 5 1 7 0 8

(32) 優先日 平成13年5月8日(2001.5.8)

(33) 優先権主張国 米国 (U S)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 インドラディーブ ゴッシュ

アメリカ合衆国, カリフォルニア 95131,
サン ノゼ, リードラー サークル 1672

(74) 代理人 100070150

弁理士 伊東 忠彦 (外1名)

Fターム(参考) 2G132 AA00 AB01 AE23 AC15 AL09

5B046 AA08 BA03 BA09 JA01

5B048 AA01 DD05

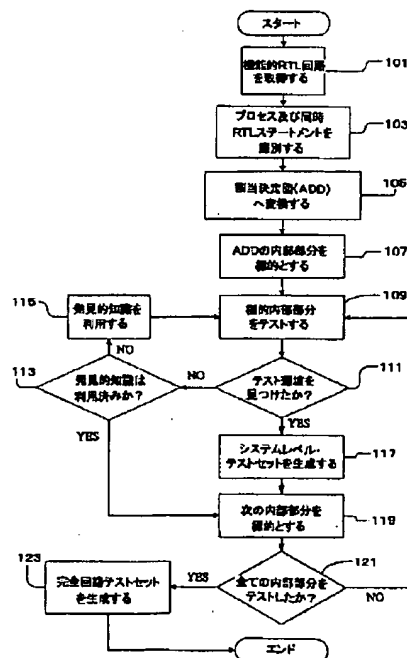
(54) 【発明の名称】 機能的RTL回路用の自動テストパターン生成方法

(57) 【要約】

【課題】 本発明は、RTLデジタル回路のテストパターンを高速かつ高品質に自動生成する方法の提供を目的とする。

【解決手段】 テストパターンはRTLデジタル回路を表現する割当決定図を作成することにより生成される。9値記号代数が使用され、目標が割当決定図内で識別された部分毎に決定される。目標は、テスト環境が見つめられた割当決定図をトラバースすることにより正当化、伝播される。テスト環境が予め見つからない場合には、発見的手法を利用する。見つめられたテスト環境を使用して、所定のテストベクトルがシステムレベル・テストセットを獲得するため伝播させられる。各部分に対するテストセットは連結され、RTLデジタル回路用の完全テストセットが得られる。

本発明によるテストパターン生成方法の一例のフローチャート



【特許請求の範囲】

【請求項1】 RTLデジタル回路の割当決定図を生成する手順と、

割当決定図内のモジュールを識別する手順と、

9値記号代数を使用して、識別されたモジュールに対する目標を決定する手順と、

テスト環境を獲得するため、割当決定図をトラバースすることにより、目標を正当化し伝播させる手順と、

獲得されたテスト環境を使用して、所定のテストベクトルをRTL回路に適用する手順と、を有するRTLデジタル回路をテストするテストパターン生成方法。

【請求項2】 テスト環境を獲得するため発見的手法を利用する手順を更に有する、請求項1記載のテストパターン生成方法。

【請求項3】 モジュールを識別し、目標を決定し、目標を正当化し伝播させ、識別された各モジュールに対しシステムレベル・テストセットを獲得するため、獲得されたテスト環境毎に識別された各モジュールに所定のテストベクトルを適用することを継続する手順を更に有する、請求項1記載のテストパターン生成方法。

【請求項4】 RTLデジタル回路用の完全テストセットを獲得するため、識別されたモジュール毎にシステムレベル・テストセットを連結する手順を更に有する、請求項3記載のテストパターン生成方法。

【請求項5】 単一クロック線を有するデジタル回路の機能的RTLレベル回路設計情報を取得する手順と、取得した回路設計情報からデータ構造を生成する手順と、

生成されたデータ構造から論理構造部及びブロックを識別する手順と、

正当化パスを識別するため、論理構造部及びブロックの入力からRTLデジタル回路の主入力まで、テスト目標を正当化する手順と、

伝播パスを識別するため、論理構造部及びブロックの出力からRTLデジタル回路の主出力までテスト目標を伝播させる手順と、

識別された正当化パス及び伝播パスを用いて、所定のテストベクトルをデジタル回路に適用する手順と、を有する、RTLデジタル回路をテストするテストパターン生成方法。

【請求項6】 テスト目標を正当化する手順とテスト目標を伝播させる手順は、9値記号代数を用いて決定された目標を使用する手順を含む、請求項5記載のテストパターン生成方法。

【請求項7】 論理構造部及びブロックを識別し、テスト目標を正当化し、テスト目標を伝播させ、識別された論理構造部及びブロック毎にシステムレベル・テストセットを獲得するため、識別された正当化パス及び伝播パス毎に識別された論理構造部及びブロックに対し所定のテストベクトルを適用することを継続する手順を更に有

する、請求項5又は6記載のテストパターン生成方法。

【請求項8】 RTLデジタル回路用の完全テストセットを獲得するため、識別された論理構造部及びブロック毎にシステムレベル・テストセットを連結する手順を更に有する、請求項7記載のテストパターン生成方法。

【請求項9】 RTLデジタル回路の割当決定図を生成する機能と、

割当決定図内のモジュールを識別する機能と、

9値記号代数を使用して、識別されたモジュールに対する目標を決定する機能と、

テスト環境を獲得するため、割当決定図をトラバースすることにより、目標を正当化し伝播させる機能と、

獲得されたテスト環境を使用して、所定のテストベクトルをRTL回路に適用する機能と、をコンピュータに実行させるためのRTLデジタル回路用のテストパターン生成プログラム。

【請求項10】 単一クロック線を有するデジタル回路の機能的RTLレベル回路設計情報を取得する機能と、取得した回路設計情報からデータ構造を生成する機能と、

生成されたデータ構造から論理構造部及びブロックを識別する機能と、

正当化パスを識別するため、論理構造部及びブロックの入力からRTLデジタル回路の主入力まで、テスト目標を正当化する機能と、

伝播パスを識別するため、論理構造部及びブロックの出力からRTLデジタル回路の主出力までテスト目標を伝播させる機能と、

識別された正当化パス及び伝播パスを用いて、所定のテストベクトルをデジタル回路に適用する機能と、をコンピュータに実行させるためのRTLデジタル回路用のテストパターン生成プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、デジタル回路用のテストパターン生成に係り、特に、割当決定図及び9値記号代数を用いて機能的なRTL（レジスタ・トランスファ・レベル）回路用のテストパターン生成に関する。

【0002】

【従来の技術】本質的に順次的であるデジタル回路用のパターン生成は、困難である場合が多い。典型的に、論理レベルで動かなくなるスタック・アット障害の検出を標的とするテストパターンを生成するためには、たとえ、適当なサイズのデジタル回路の場合であっても、大量の計算時間及びリソースが必要である。回路を変更することによりデジタル回路の複雑さを低減するテスト可能用設計（DFT）技術が提案されている。しかし、回路のテスト可能性が増大しても、領域オーバーヘッドの不利、性能の不利、電力オーバーヘッドの不利、或い

は、これらの組み合わせの問題が屢々生じる。したがって、DFT技術を使用することは、必ずしもテストパターンをより容易に生成するための選択肢ではない。

【0003】さらに、従来のテストパターン生成器は、屢々、デジタル回路の論理レベルを標的としている。論理レベルでは、スタック・アット障害を適切に取り扱う大規模順序回路用のテストパターンを生成することは不可能である。たとえ、テストパターンが大きな困難を伴って生成されたとしても、論理レベルで発見される問題は、設計プロセスの簡単化や、設計所要時間の短縮に貢献しない。

【0004】

【発明が解決しようとする課題】RTLの場合、RTLでのデジタル回路における基本素子の数が論理レベルよりも減少するので、サイズに関する問題は緩和される。その結果として、テスト生成問題は簡単化される。同時に、設計プロセスは、屢々、RTLで始まるので、誤りの発見は論理レベルよりも容易に確定される。

【0005】本発明は、RTL回路用のテストパターン生成システム及び方法の提供を目的とする。

【0006】

【課題を解決するための手段】本発明の一局面によれば、テストパターンは、RTLデジタル回路の割当決定図を生成することによって生成される。割当決定図から、モジュールが識別され、目標が9値記号代数を用いて判定される。目標は、テスト環境を得るため、割当決定図をトラバースすることによって正当化され伝播される。所定のテストベクトルは、獲得されたテスト環境を使用して、RTLデジタル回路内に伝播させられる。本発明の一局面において、割当決定図は、RTLデジタル回路の機能的挙動を表現するグラフ的表現と、RTLデジタル回路の近似的構造表現とを有する。本発明の更なる局面において、識別された一つのモジュールに対し予め計算されたテストベクトルは、識別されたモジュールに対するシステムレベル・テストセットを獲得するためテスト環境に適用され、識別されたモジュール毎のシステムレベル・テストセットは、RTLデジタル回路用の完全なテストセットを獲得するため連結される。

【0007】本発明の他の局面において、テストパターンは、単一クロック線を有するデジタル回路の機能的RTL回路設計を取得することにより生成される。取得された回路設計からのデータ構造、及び、データ構造からのブロックが識別される。テスト目標は、正当化パスを識別するため、論理構造の入力と、RTLデジタル回路の主入力からのブロックとにより正当化される。所定のテストベクトルは、識別された正当化パス及び伝播パスを用いて、デジタル回路内に伝播させられる。本発明の更なる局面によれば、識別された一つの論理構造及びブロックに対し予め計算されたテストベクトルは、識別された一つの論理構造及びブロックに対するシステムレベ

ル・テストセットを獲得するため、正当化パス及び伝播パスに適用され、識別された一つの論理構造及びブロックに対するシステムレベル・テストセットは、RTLデジタル回路用の完全テストセットを得るため連結される。

【0008】本発明の多数の付随的な特徴は、添付図面を考慮して以下の詳細な説明を参照することによってより良く理解されるので、容易に認められるであろう。

【0009】

【発明の実施の形態】図1は、本発明によるテストパターン生成方法の一実施例のフローチャートである。ステップ101において、機能的RTL回路記述を取得する。ステップ103で、各プロセスと、RTL回路の各リーフ・コンポーネント内の同時RTLステートメントを識別する。ステップ105において、識別されたRTLステートメントを割当決定図(ADD)に変換する。一つのADD内部の部分若しくはモジュール、たとえば、算術モジュールを標的(ターゲット)にする。ステップ109において、標的の各部分をテストする。ADDの一部は、9値記号RTL正当化及び伝播法を用いてテストする。RTL正当化及び伝播法は、回路の主入力から内部の入力までのパスと、内部の出力から主出力までのパスを追跡する。正当化及び伝播法で使用される9値記号代数については、後で詳述する。

【0010】一実施例において、正当化及び伝播は、バックトラック及び探索時間制限を調節して利用することができるバックトラッキングによるブランチ・アンド・バウンド型の探索である。ステップ101で、階層型RTL回路記述を取得する場合、階層型探索トラバースを利用する。演算における全ての変換は、9値代数に基礎を置き、ルックアップテーブルに格納されている。ステップ111において、内部の部分のテストの結果としてテスト環境が得られたかどうかを判定する。ステップ111において、テスト環境が見つからなかったと判定した場合、ステップ113に進み、発見的知識が予め利用されたかどうかを判定する。発見的知識が利用されていないと判定した場合、ステップ115において、発見的知識を利用する。発見的知識を利用することにより、内部の部分のテスト範囲が広がる。そして、ステップ109へ戻り、ステップ115で与えられた発見的知識を利用してADDの内部の部分のテストを継続する。

【0011】ステップ113において発見的知識を既に利用したと判定され、かつ、ステップ111においてテスト環境が見つからなかった場合、標的とする内部の部分に対しテスト環境を決定できないと判定する。このとき、ステップ119へ進み、次の内部の部分の部分を標的とする。

【0012】ステップ111において、テスト環境が決定したと判定された場合、ステップ117で、その内部の部分に対するシステムレベル・テストセットを生成す

る。ステップ119において、ADD内部の次の部分、たとえば、他の算術モジュールを標的とする。処理は、ステップ121でADDの全ての内部部分がテストされたと判定されるまで、ADDの他の部分と、全てのADD（論理アレイ、記憶ユニット、内部接続、ランダム論理ブロック、及び、ブラックボックス）をテストし続ける。（ステップ115で）テストした全ての内部部分に対し生成されたシステムレベル・テストセットは、ステップ123で、完全RTL回路テストセットを生成するため連結され、一連の処理が終了する。

【0013】ステップ105では、RTLステートメントをADDに変換する。図2には、このような割当決定図が示されている。デジタル回路は、主入力と、主出力と、内部記憶素子とを有する。デジタルシステムは、主入力へ与えられた値に関する計算のセットであり、その計算の結果は内部記憶素子に格納されるか、或いは、主出力へ渡される。かくして、標的がデジタルシステムの記憶素子若しくは出力ポートを表す場合に、割当決定図は、一つ以上の標的への条件付き割当のセットとして、デジタル回路を表現する。

【0014】割当決定図は、一方向性の鎖状グラフである。割当決定図は、割当値セクション21と、割当条件セクション23と、割当決定セクション25と、割当標的セクション27の4つのセクションを含む。各セクションは、一つ以上のノード、読み出しノード、書き込みノード、演算ノード、又は、割当決定ノードを含む。各ノードは、異なる属性を有し、他のノードへ接続される。ノード間の接続は、一方向性エッジによって表現される。

【0015】割当値セクション21は、最終的に記憶素子又は出力ポートへ割り当てられる値の計算を表現する。各値は、入力ポートの現在値、記憶素子、又は、定数から計算され、読み出しノード、たとえば、読み出しノードR1及びR2として表現される。計算は、たとえば、論理演算若しくは算術演算の演算ノードO1及びO2を有するデータフローグラフとして表現される。

【0016】割当条件セクション23は、演算ノードO3〜O7に接続された読み出しノードR3〜R6を含む。読み出しノードR5は、多状態設計用の制御手順シーケンサとして動作する記憶ユニットを表現する。換言すると、状態レジスタは、割当条件によって順序付けられた一連の制御命令又は割当値を含む。計算の最終結果AS1及びAS2は、ブール代数值であり、割当条件選択信号を表現する。最終結果、又は、割当選択信号AS1及びAS2は、割当値セクション21のための保護条件として作用する。

【0017】割当決定セクション25は、割当決定セクションの入力へ与えられた値のセットから値を選択する割当決定ノードAD1を含む。割当入力値は割当値セクションから与えられ、選択入力値は割当条件セクションから

与えられる。かくして、割当選択信号AS1が真であるならば、割当入力AI1が選択される。割当選択信号AS2が真であるならば、割当入力AI2が選択される。選択入力AS1及び選択入力AS2が真ではない場合、割当入力AI1とAI2はどちらも選択されない。入力AI1とAI2の中の一つだけが同時に選択できるので、選択入力AS1とAS2は、同時に真であると評価されることは無い。

【0018】割当標的セクション27は、書き込みノードを含む。本実施例では、割当標的セクションは書き込みノードWI1を含む。書き込みノードは、割当決定セクションから値を取得する。すなわち、書き込みノードWI1は、割当決定ノードAD1から選択された値を取得する。また、ある時点において、割当決定ノードへの割当入力の一つだけが真であると評価されるので、同時に割当標的に割り当てられる値は1個だけである。かくして、割当標的に対する全ての割当条件は、相互に排他的である。

【0019】一実施例において、多数の書き込みポート（図示されない）を備えた記憶ユニットは、多数の書き込みポートの中の一つの書き込みポートへの割当を表現する多数の割当決定ノードによって表現される。割当値は、対応した割当条件が真であると評価された場合に限り、割当決定ノードを介して書き込みポートへ書き込まれる。多数の読み出し／書き込みポート（図示されない）を備えた記憶ユニットは、それぞれの読み出し／書き込みポートに割り当てられたインデックスを含む2次元アレイによって表現される。インデックスは、割当値が読み出された場所、又は、割当値が書き込まれる場所を識別する。多数の読み出しポートを備えた記憶ユニットは、多数の出力線を有する読み出しノード、たとえば、読み出しノードR1及びR2のような読み出しノードとして表現される。

【0020】一実施例において、割当決定図のサイズは、全てのポート及び記憶ユニットに対する条件付き割当の数に比例する。割当決定図は、矛盾の無い体系的なデータフロー形式で条件及び計算を表現する。そのため、割当決定図は、スタック・アット障害を容易に見つける具体的な総合的決定論に依存することなく、最終的な構造的ネットリストと類似する。また、割当決定図によって表現されるように制御とデータが統合されるので、制御バスとデータバスの分離問題はテスト中に回避される。

【0021】再度図1、特に、図1のステップ101を参照すると、RTL回路記述が取得される。図3には、ハードウェア記述言語として表現された機能的RTL回路が示されている。VHDL又はVerilogのようなハードウェア記述言語から、各プロセスは割当決定図に変換される。有限状態機械（FSM）の組合せプロセス及び順序プロセスの場合、二つのプロセスが一つの割

当決定図に合成される。図4には、図3のRTL回路記述の割当決定図が示されている。たとえば、RTLステートメント301(図3)は、読み出しノードA及びBと、演算ノード63と、割当決定ノード41と、書き込みノードNEXT_Fとにより表現される。図4において、RESET、APORT、BPORT、CPORT、DPORT及びEは、主入力表現し、OPORTは主出力を表現する。NEXT_A、NEXT_B、NEXT_C、NEXT_D、NEXT_F、NEXT_G、NEXT_O及びNEXT_STATEは、割当標的である。CURRENT_STATEは、記憶ユニットである状態レジスタを表現する。

【0022】割当値A、B、C、D、E、F、O、G、3及びOは、読み出しノードとして表現される。さらに、APORT、BPORT、CPORT、DPORT、S0、S1、S2及びS3は、読み出しノードとして表現された割当値である。多数の演算ノードが一部の読み出しノードに接続され、たとえば、A+B、A-Bなどの計算を表現する。これらの読み出しノード及び演算ノードは、割当決定図の割当値セクションである。割当値セクションは、割当決定ノード31、33、35、37、39、41、43及び45を含む割当決定セクションへ接続される。割当決定セクションへ選択入力を供給する割当条件セクションは、読み出しノードS0、S1、S2、S3及びRESETと、読み出しノードに接続された種々の演算ノードとを含む。

【0023】状態遷移図は、図5に示されるように、たとえば、Current_STATE状態変数に対し生成される。入力線をセット又はリセットすることにより達成される状態は、リセット状態である。有限状態機械が存在するプロセスへの入力であるRTL変数からデータが読み出される状態は、入力状態である。図5において、リセット状態は状態S0であり、入力状態は状態S1である。出力状態は状態S3である。出力状態は、有限状態機械内の計算の結果が、有限状態機械の存在するプロセスへの出力であるRTL変数に書き込まれる状態である。

【0024】図1、特に、図1のステップ109を参照するに、一実施例において、9値記号代数は、RTL回路の記号的正当化及び伝播を実行するため利用される。9個の記号には、汎用可制御性Cg、零への可制御性C0、1への可制御性C1、全て1への可制御性Ca1、定数への可制御性Cq、z値への可制御性Cz、状態Csへの可制御性、可観測性O、及び、可観測性のコンプリメントO'が含まれる。類似した記号代数が、構造的デジタル回路設計の際に、内蔵型自己テストスキームのため使用される。このようなスキームの一例は、参考のため引用される文献: Ghosh et al., "A BIST Scheme for RTL Circuits Based on Symbolic Testability Analysis"に記載されている。

【0025】RTL変数の汎用可制御性記号は、記号の値を任意の値へ制御できる記号の能力を表現する。たとえば、特定のRTL変数に関して決定された汎用可制御性記号のバイト長がnである場合、RTL変数は、RTL回路の主入力からRTL変数に関して実現可能な 2^n 個の値へ制御可能である。

【0026】RTL変数の記号0への可制御性は、RTL変数を値0へ制御する能力を表現する。RTL変数の記号1への可制御性は、RTL変数を値1へ制御する能力を表現する。RTL変数の全て1の記号への可制御性は、RTL変数を全て1の値へ制御する能力を表現する。たとえば、多数バイト変数の場合に、全てのビットは値1を有する。これに対し、記号1への可制御性の場合、多数バイト変数に対し一つのビットだけが値1をとる。しかし、全て1の記号への可制御性は、単一ビットのRTL変数の場合には、記号1への可制御性と同じである。2の補数演算の場合、全て1への可制御性は、値-1への可制御性に対応する。

【0027】RTL変数の定数記号への可制御性は、RTL変数を固定値へ制御する能力を表現する。たとえば、nバイトのRTL変数の場合、定数記号への可制御性は、RTL変数を 2^n 個の候補値の中の一つの定数へ制御する能力を表現する。単一ビット変数の場合、定数記号への可制御性は、RTL変数を0又は1へ制御する能力を表現する。RTL変数の値zの記号への可制御性は、RTL変数を高インピーダンスz状態へ制御する能力を表現する。有限状態機械で定義された状態変数の状態記号への可制御性は、状態変数を特定の状態へ制御する能力を表現する。RTL変数の可観測性記号は、RTL変数の障害を観測する能力を表現する。たとえば、多数バイト型RTL変数の場合、可観測性記号は、正しい値若しくは予測値とは異なる誤りを含む値がRTL変数上に存在することを表す。コンプリメント可観測性記号は、単一ビット型RTL変数だけに適用可能であり、0/1障害、すなわち、RTL変数に関する予想値が0であり、障害値が1であることを表す。

【0028】RTL変数に関する汎用化制御性記号は、同じRTL変数に対する、零への可制御性、1への可制御性、全ての1への可制御性、及び、定数記号への可制御性を表現する。同様に、零への可制御性、1への可制御性、又は、全て1の記号への可制御性は、同じRTL変数に対する定数記号への制御可能性を表現する。

【0029】9個の記号を使用することにより、正当化及び伝播のための変形規則が定式化される。たとえば、加算演算の出力における汎用可制御性は、加算演算の一方の入力における汎用可制御性記号と、加算演算の他方の入力における定数記号へ可制御性へと変形される。同様に、乗算演算の一方の入力側の可観測性記号は、乗算演算の出力側での可観測記号と、乗算演算の他方の入力側での1への可制御性若しくは全ての1の記号への可

御性とに変形される。一実施例において、変形規則は、ルックアップテーブルに格納される。

【0030】デジタル回路には、屢々、分離若しくは併合するバスが含まれるので、ビットインデックスをRTL変数に付加する。また、デジタル回路は本質的に順序性があるので、時間フレームを9値記号代数に導入する。かくして、9値記号代数を使用することにより、RTL正当化/伝播レコードが、ATPG目標、変数名、変数の上位ビットインデックス及び下位ビットインデックス、時間フレーム値、並びに、状態値を含む6個の値のセットとして定義される。たとえば、Cg(a[7:0], 0, -)は、時間フレーム0内のRTL変数a[7:0]に関する汎用可制御性の目標が満たされるべきであることを表現する。同様に、Cs(Current_state, 3, S0)は、状態変数Current_stateの時間フレーム3における状態S0への可制御性が満たされるべきことを表現する。

【0031】このように、9値記号代数を使用することにより、RTL回路の正当化及び伝播バスが記述される。たとえば、プロセスは、割当決定図の内部部分、すなわち、演算ノード63のような算術モジュール、本例の場合には、状態S1における加算演算を標的とする。加算演算用の演算ノード63の読み出しノードA及びBの二つの入力の制御が必要である。また、書き込みノードNEXT_Fの観測と、有限状態機械のトラバースとが、正当化及び伝播の目標を満たすために必要である。図5を参照するに、有限状態機械は、リセット状態である状態S0から、入力状態である状態S1と出力状態である状態S3へトラバースされる。

【0032】図6は、割当決定図の内部部分の正当化及び伝播探索を実行するサブプロセスの一実施例を示すフローチャートである。ステップ201において、初期目標が標的となる内部部分、たとえば、加算演算ノードに対し設定される。たとえば、加算演算ノード63に対し、初期目標は、加算演算の出力の可観測性と、加算演算への二つの入力の汎用可制御性である。加算演算は状態S1で行われるので、CS目標、すなわち、状態への可制御性が含まれる。ステップ203において、第1の目標は、ADDによって記載されるRTLモジュールの機能性に基づいて割当決定図の中で伝播させられる。ステップ205において、主入力に達したかどうかを判定する。ステップ205において、主入力に到達したと判定した場合、目標が満たされたと考えられるので、標的とするADDの内部部分に対するテスト環境を表現するテーブルに目標を格納する。ステップ211において、目標の全てが満たされたかどうかを判定し、目標の全てが満たされたと判定した場合、このサブプロセスを終了し、呼出元へ返る。満たされるべき多数の目標が残っていると判定した場合、ステップ209において、次の目標を獲得し、ステップ203で新しい目標を伝播させ

る。

【0033】ステップ205において、主入力に達していないと判定された場合、ステップ213で目標を変形する。一実施例において、目標は、ルックアップテーブル内の所定のレコードに基づいて変形される。また、必要に応じて、目標の時間フレームがインクリメント若しくはデクリメントされる。ステップ215において、新しい目標が、たとえば、既に満たされた目標のような他の目標と競合するかどうかを判定する。新しい目標が他の目標と競合しないと判定した場合、ステップ203において、新しい目標を使用してADDをトラバースすることにより、サブプロセスが繰り返される。しかし、ステップ215において、競合が存在すると判定された場合、ステップ217でバックトラックを行う。探索が一方向で失敗した場合、探索はバックトラックを行い、中間的な決定結果を廃棄し、新しい方向で継続する。一実施例において、バックトラック限界及び/又は時間制限が探索毎に設定される。処理は、再びステップ213へ戻り、新しい目標を生成し、ステップ215において、新しい目標が他の目標と競合するかどうかを判定する。伝播目標は、同様に決定され、サブプロセスによって満たされる。但し、ステップ205において、目標が満たされたかどうかを判定するため使用する条件は、主入力ではなく、主出力である。

【0034】図7は、図6を参照して説明した算術モジュールをテストする正当化及び伝播探索サブプロセスで記憶され操作される目標を示すブロック図である。初期目標はスタック511に入れられる。RTLモジュールの機能性に応じて、目標は、スタックの上から変形される。制御論理のトラバース中の中間手順の個数は膨大であるので、ある種の中間手順は図示されず、図面を見易くするため、マーク553で示される。マーク553は、多数の選択範囲を示す。なぜならば、変形及びバックトラックのための一つ以上の選択範囲がその点で必要になるからである。正当化のため、シフト中の目標及び満たされた目標を収容するスタックの第1行に示されるように、状態シーケンスが決定される。加算演算への演算入力から種入力までの正当化バスは、スタック551e~551fに示されるように決定される。図5に示されるように状態遷移図に従う伝播バスは、スタック551g~551mに示されるように決定される。

【0035】次に、図4乃至7を参照して、正当化目標を解決するための割当決定図のトラバースを詳述する。読み出しノードに到達し、読み出しノードが主入力若しくはプロセス、すなわち、現在状態への入力では無い場合に、割当決定図がトラバースされ、目標は対応した書き込みモードへ移される。たとえば、変数B、すなわち、読み出しノードBの汎用可制御性目標が対応した書き込みモードNEXT_Bへ移される。しかし、読み出しノードに到達し、読み出しノードが主入力である場

11

合、目標は満たされる。また、到達した読み出しノードがプロセス入力である場合、目標は、別のプロセスの書き込みノード、又は、現在プロセスへ連結した同時的なRTLステートメントへ移される。

【0036】或いは、書き込みノードに到達したとき、対応した目標は、到達した書き込みノードへ接続された割当決定ノードの一つの割当値入力へ移され、一つの目標への可制御性が割当決定図への対応した条件入力に要求される。正当化パスの選択肢の数は、割当決定ノードに接続された入力の数に基づいている。こうして、スタック551a〜551bに示されるようにバックトラックが必要になる。

【0037】クロックエッジ・テスト用表現サブツリー、すなわち、サブツリー331が存在し、その表現サブツリー内の重み又はイベントステートメントが割当決定ノードに条件入力を供給する場合、目標の時間フレーム数は、スタック551b及び551dに示されるように1ずつデクリメントされる。演算ノードの出力、すなわち、出力51(図3)に到達したとき、出力は、演算の機能性及びRTL代数に従って変形される。異なるRTL及び論理演算に対する変形は、ルックアップテーブルに格納される。伝播パスの決定は、上述の正当化パスの決定と同様に行われるが、主出力から演算出力の方へ逆向きに行われる。また、割当決定ノードの制御入力から割当決定ノードの出力までの伝播中に、割当決定ノードの対応した割当値入力と相補的条件値入力は、異なる値をとる必要がある。

【0038】新たに割り当てられた目標は、既に満たされた目標と競合する場合がある。たとえば、同じ時間フレーム内の同じ変数の汎用可制御性が既に決定されたとき、汎用可制御性が特定の時間フレーム中の変数に要求される場合、競合が生じるので、同じ時間フレーム内で変数を二つの異なる任意の値へ制御することは不可能である。同様に、変数の零への可制御性は、同じ変数に対する1への可制御性と競合する。かくして、満たされた目標のリストが保持され、競合する新しい目標がスタック内に入れられないように調べられる。

【0039】正当化及び伝播中に、時間フレーム数は、初期目標上の0から始まり、正当化中にデクリメントして負の大きい数になり、伝播中にインクリメントして正の大きい数になる。正当化及び伝播中に、正当化最前線が満たされるべき目標と共にスタックに保持される。正当化最前線は、出力値が既知であるゲートのセットを表現する目標のセットであるが、ゲートの出力値は入力値からは示唆されない。また、伝播中に、0-最前線は維持される。0-最前線は、任意である一つの出力値を有するゲートのセットを表現する目標のセットである。しかし、ゲートへの入力側には一つ以上の誤差値がある。

【0040】正当化パス及び伝播パスのセットが標的となる内部部分、すなわち、加算演算に対し決定された

12

後、加算演算用のテスト環境が見つけられる。図8は、システムレベル・テストを生成するためテスト環境を使用するサブプロセスの一実施例のフローチャートである。ステップ301において、正当化／伝播サブプロセスで見つかった正当化／伝播レコード内の目標の非常に大きい負の時間フレームを零化し、他のレコード内の他の時間フレームを更新する。たとえば、非常に大きい負の時間フレーム、時間フレーム-2には、零が設定され、他の時間フレーム番号は、それぞれ更新され、たとえば、-1は1に更新され、0は2に更新され、1は3に更新され、以下同様である。ステップ303において、最大時間フレーム、すなわち、ステップ301で更新された時間フレームの中で最大時間フレーム値を設定する。加算演算用に予め計算されたテストセットは、ステップ305でテストベクトルをテスト環境に適用するため使用される。たとえば、加算演算の場合に、値0及び値5が、テストベクトルとして、加算演算の左側入力及び右側入力にそれぞれ必要となる場合に、読み出しノードAは零が割り当てられ、読み出しノードBは5が割り当てられる。次に、主入力への値を正当化するためテスト環境が使用され、主入力A PORTで値0を取得し、主入力B PORTで値5を取得する。使用されない主入力にはランダムな値が与えられる。

【0041】ステップ307において、指定された時間フレームに対する正当化／伝播レコードで目標に応じたテストベクトルを伝播させる。ステップ309で、主入力での値を取得し、ステップ311で、テストシーケンスを保存する。使用されない主入力にはランダム値が割り当てられる。つぎに、サブプロセスは終了し、呼出側へ戻る。このサブプロセスは、テストベクトルの数に対し必要な回数だけ繰り返される。予め計算されたテストセットのテストベクトルから獲得された全てのテストシーケンスは、一つに連結され、加算演算のためのシステムレベル・テストセットが生成される。テストシーケンスの長さは、テスト環境内の最大時間フレーム番号に対応する。

【0042】一実施例において、算術演算のタイプ毎に予め計算されたテストセットは、実現可能なバイト幅毎にライブラリに収容される。予め計算されたテストセットは、一実施例の場合、算術演算の直列実装と、同じ算術演算の並列実装の組合せに対するテストセットから決定される。一実施例において、利用可能であるならば、実装に依存しないテストセットが、予め計算されたテストセット、又は、二つの組合せの代わりに使用される。

【0043】図1を参照するに、算術モジュールを標的とし、テストした後、論理アレイを標的とする。論理アレイは、上述の算術モジュールと同様にテストされる。しかし、多重バイト型論理演算若しくは論理アレイを標的とし、テストするときには、テストセットが常に一定の範囲内に収まることを確認する。汎用可制御性目標を

使用することなく、各テストベクトルを個別に正当化し、伝播させる。たとえば、左右の入力バスを備えたORゲートアレイの場合、テスト環境は、{(Ca1, C0), (C0, Ca1), (C0, C0)}である。

【0044】レジスタ、ラッチ及びメモリのような記憶素子を標的とし、テストする。図9には、記憶素子をテストするサブプロセスの一実施例のフローチャートが示されている。ステップ401で、記憶素子が既にテスト済みであるかどうかを判定する。一実施例の場合、記憶素子が既にテスト済みであるかどうかを判定するため、10 標的とされる現在の記憶素子をテスト済みのモジュールのリストと比較する。記憶素子がテスト済みである場合、このサブプロセスは終了する。算術モジュール及び論理モジュールのテスト中に、多数のレジスタ及びラッチがテストされる。したがって、これらのレジスタ及びラッチは再テストしない。

【0045】一実施例において、他の内部部分、たとえば、他の算術モジュールのテスト環境の決定中に、Cg目標が書き込みノード、読み出しノード又は割当決定ノード値ポートの変数に対し満たされた場合、書き込みノード、読み出しノード、及び、割当決定ノードの値ポートには、タグが付けられ、たとえば、1でマークされる。かくして、書き込みノード、読み出しノード、又は、割当決定ノードの値ポートにタグが付いていると判定した場合、そのノード又はポートのテストは省略する。

【0046】ステップ401において、記憶素子は未だテストしていないことが判明した場合、ステップ403で、記憶素子がメモリユニットであるかどうかを判定する。記憶素子は、割当決定図の観点からは、割当決定ノードの出力に接続された書き込みノードである。割当決定ノードへの一つの条件入力クロックエッジ型テスト用サブツリーに接続されているならば、記憶素子はレジスタとして実装され、そうでなければ、記憶素子はラッチとして実装される。メモリは、典型的に、RTL回路記述愛では配列宣言によって形成される。

【0047】記憶素子はメモリユニットであることが判明した場合、ステップ405において記憶素子用のテスト環境を生成し、記憶素子をテストするため、ステップ407で所定のテストパターンをテストベクトルとして 40 ロードする。ここで、サブプロセスは終了する。たとえば、全ての未だテストされていない記憶素子は、各記憶素子に0パターン及び1パターンをロードすることによってテストし、その出力を観測する。このようにして、一つのテスト環境は、零目標への可制御性を用いて獲得され、別のテスト環境は、全て1への可制御性を用いて獲得される。

【0048】記憶素子はメモリユニットであることが判明した場合、ステップ409でメモリユニットに対するテスト環境を生成する。一実施例において、テスト環境 50

は、データ線及びアドレス線又は変数でのメモリに対する汎用可制御性目標を、右側制御ポート上の目標1への可制御性、又は、制御ポートが負論理であるならば、目標0への可制御性と共に使用して見つけられる。つぎに、別のテスト環境が、メモリのアドレス変数での汎用可制御性目標と、読み出しポート側での目標1への可制御性(負論理の場合には、零への可制御性)と、メモリの出力ポート側での可観測性目標とを使用して見つけられる。ステップ411では、テストパターンをメモリにロードし、パターンを読み出すことにより、メモリユニットのテスト環境をテストする。ここで、サブプロセスは終了する。サブプロセスは、標的とする記憶素子毎に繰り返される。

【0049】本発明のテストパターン生成方法の一実施例によれば、図1に示されるように、内部接続を標的とし、テストする。回路の内部接続は、バス又はマルチプレクサのいずれかの形式である。たとえば、2入力マルチプレクサは、入力による二つの割当とコンプリメント条件入力を含む決定ノードとして表現される。本例の場合、割当決定ノードへの割当値入力はV1及びV2であり、条件入力はSであり、Sが値1をとるならば、入力V1が選択され、Sが値0をとるならば、入力V2が選択される。したがって、マルチプレクサの完全なテストを可能にするためには、4種類のテスト環境{[Ca1(V1), C1(S)], [Ca1(V2), C0(S)], [C0(V1), Ca1(V2), C1(S)]及び[Ca1(V1), C0(V2), C0(S)]}が必要になる。レジスタの場合、レジスタロードは、典型的に、レジスタ変数を表現する書き込みノードの上部に2入力割当決定ノードとして表現される。これにより、割当決定ノードをテストすると、レジスタロード線もテストできる。

【0050】複雑な多入力ADNノードをテストするため、テスト環境が、ADNノードへの各値入力に関するCa1必要条件と、対応した条件入力に関するC1必要条件とを用いて見つけられる。たとえば、5個の条件入力S1、S2、S3、S4及びS5と、値入力V1、V2、V3、V4及びV5とを備えた複雑な多入力ADNノードの場合、テスト環境は以下の目標：{[Ca1(V1), C1(S1)], [Ca1(C2), C1(S2)], [Ca1(V3), C1(S3)], [Ca1(V4), C1(S4)], [Ca1(V5), C1(S5)]}を用いて見つけられる。一実施例において、ADNは、多入力マルチプレクサ・ツリーに合成される。マルチプレクサ・ツリーへの入力の順序は不明であるため、選択線スタック・アット・テストだけが、一つの値入力におけるC0の条件、たとえば、C0(V1)と、対応した条件入力におけるC1の条件、たとえば、C1(S1)の下で行われる。したがって、5入力ADNノードを使用する本例の場合、テスト環境を見つ

けるため、以下の目標：{ [C0 (V1), C1 (S1)], [C0 (C2), C1 (S2)], [C0 (V3), C1 (S3)], [C0 (V4), C1 (S4)], [C0 (V5), C1 (S5)] } を使用してもよい。

【0051】さらに、本発明の一実施例による方法は、ランダム論理ブロックを標的とし、テストする。ランダム論理ブロックは、割当決定ノードへの条件ポートを制御する論理として表現される。ある値が割当決定ノードへの入力し、割当決定ノードが一定のままである場合、割当決定ノードはランダム論理ブロックとして合成される。ランダム論理割当決定ノードは、このように識別され、割当決定ノードへの各条件入力信号を1に制御し、割当決定ノードの出力の書き込みノードの要求を満たすことによりテストされる。

【0052】一実施例において、テスト中のRTL回路は、たとえば、RTL設計情報が公開されていないブラックボックスと呼ばれるコンポーネントを含む。このため、付加情報が与えられない限り、ブラックボックスの境界を通過する正当化及び伝播パスを決定できない。要求される付加情報には、ブラックボックスへの各入力がある一定のサイクル数の範囲内で一つ以上の出力へ伝播する様態を識別する記述が含まれる。維持された0-最前線は、付加情報を用いてブラックボックスの向こう側へ伝播させられる。一実施例において、要求される付加情報には、ブラックボックスの各出力がある一定のサイクル数の範囲内で一入力、又は、入力の組合せから正当化される態様が含まれる。しかし、ブラックボックスは、本質的に組合せ的であり、テストベンチを利用することができるので、ブラックボックスの入力側の汎用可制御性標的及びブラックボックスの出力側の可観測性を用いてテスト環境を見つけることにより、ブラックボックスを算術演算としてテストすることが可能である。ブラックボックスが順序的である場合には、テストベクトルを正当化及び伝播させるために多数のサイクルが必要となり、付加的なテストビリティ用設計装置又は技術が必要である。

【0053】図10を参照するに、標的とするブラックボックスをテストするサブプロセスの一実施例のフローチャートが示されている。ステップ601において、所*

* 定の正当化及び伝播パスを、変形のため利用されるルックアップテーブルに追加する。ステップ603において、所定のテストセットをテストセットライブラリに追加する。ステップ605において、ブラックボックスが順序的であるかどうかを判定する。ブラックボックスが順序的であると判定された場合、ステップ607で、テスト用設計(DFIT)装置が利用され、主入力から内部部分の入力へ、並びに、内部部分の出力から主出力へのテストベクトルの正当化及び伝播中に内部部分に供給されたクロックを操作する。ステップ609において、所定の正当化及び伝播パス、並びに、テストセットを使用してブラックボックスをテストする。ここで、サブプロセスは終了する。

【0054】このように割当決定図の内部部分が、図1を参照して説明したように標的とされテストされるとき、他の状況も生じる。たとえば、一実施例において、バス分離及び/又は併合が起こり、このようなバスを取り扱うため、RTL代数に基づいて種々の変形規則を使用する。

【0055】すなわち、正当化及び伝播中にADDをトラバースするとき、分離ノード及び併合ノードを識別し、適切な変形を実行する。たとえば、正当化中に、バスb[1:0]を形成するため2本のビット線を併合する場合、目標C1(b[0:0])及びC1(b[1:1])をCa1(b[1:0])に変形する。同様に、伝播中に、たとえば、バスa[3:0]が4本のビット線に分離する場合、目標O(a[3:0])をO(a[0:0])とO(a[1:1])とO(a[2:2])とO(a[3:3])に変形する。別の例では、バスb[5:2]が別個のビット線に分離する場合、伝播中に、目標O(b[5:2])は、O(b[5:5])とO(b[4:4])とO(b[3:3])とO(b[2:2])に変形される。

【0056】表1には、正当化中に、1本以上のビット線が(行1と行2に)併合するとき、並びに、伝播中に、1本以上のビット線が併合するときの標的の変形の更なる例が示されている。

【0057】

【表1】

表1

事例	初期目標	新しい目標
1	Ca1(a[7:4]), Ca1(a[3:2]), Ca1(a[1:0])	Ca1(a[7:0])
2	Cq(d[3:2]), Cq(d[1:0])	Cq(d[3:0])
3	C1(c[3:2]), C0(c[1:1]), C0(c[0:0])	Cg(c[3:0])
4	O'(e[8:8]), O(e[7:6]), O(e[5:4]), O'(e[3:3])	O(e[8:3])

一部の事例では、バス目標は、全体として、伝播若しくは※50※は正当化を失敗するので、目標はビットレベル目標に関

して拡張される。たとえば、Cg目標をC1目標とC0目標のランダムセットに拡張し、次に、正当化及び伝播が行われる。

【0058】他の例の場合、非同期式セット/リセットを有するレジスタが出現する。非同期式セット/リセットを割当決定図に表現するため、レジスタに格納された変数に対応する書き込みノードは、過渡的ファンインにクロック・ツリー表現を含まない一つの制御入力に有する割当決定ノードに接続される。伝播及び正当化中に、選択された割当決定ノードへ入力された割当値が制御入力と対応する場合、時間フレームはそのまま維持され、インクリメント若しくはデクリメントされない。このように、非同期式動作が保たれる。

【0059】一実施例において、変数の正当化は繰り返しの先行時間フレーム内の同じ変数に依存し、その他の正当化の可能性は全て尽くされたと判定した場合、カウンタ変数を検出したと判定する。たとえば、正当化目標のパターンがCg(a[3:0], 0)、Cg(a[3:0], -2)、Cg(a[3:0], -4)であるならば、変数a[3:0]をカウンタ変数として認定する。カウンタ変数が認定された後、カウンタ変数のリセット線を見つけ、カウンタをリセットし、カウンタの出力を一定のサイクル数に亘って主出力へ伝播させる。

【0060】一実施例において、RTL回路は階層的である。異なるコンポーネント間の接続性を指定するトップレベル接続性ファイルを使用して、階層設計を取り扱う。正当化中に、正当化最前線がコンポーネントの入力に到達したとき、接続性は、正当化最前線を別のリーフ・コンポーネントの出力、又は、主入力へ伝達するため使用される。これにより、RTL回路の全てのリーフ・コンポーネントでテストが行われる。階層的トラバースを補助するため、目標は階層的文字列でタグ付けされる。たとえば、変数AがコンポーネントXにあり、コンポーネントXがコンポーネントYにあり、コンポーネントYがコンポーネントZにあるとき、変数Aの汎用可制御性Cg(A)は、階層的な文字列X.Y.Zでタグ付けされる。

【0061】一実施例において、テスト環境を見つけることができず、発見的手法が使用される場合がある。一実施例では、発見的部分正当化及び伝播が使用される。これにより、要求された正当化及び伝播目標の一部を満たす部分テスト環境が見つけれられる。部分テスト環境は、テストシーケンスを生成するため使用される。

【0062】伝播目標が満たされない場合、伝播発見的手法が使用され、この伝播発見的手法で、所定の数のランダムテストベクトルが正当化に基づく部分テスト環境に追加され、誤差を主出力へ伝播させる試みがなされる。

【0063】以上の通り、本発明は、RTL回路用のテストパターン生成方法及びシステムを提供する。本発明

は、特定の実施例に関して説明されているが、種々の異なる変形及び変更が当業者には容易である。したがって、本発明は、詳細に説明した実施態様以外の態様でも実施できることに注意する必要がある。また、上記の本発明の実施例は、あらゆる観点から本発明を例示するものであり、本発明を限定するものではないことが斟酌されるべきである。本発明の範囲は、請求項に記載された事項と、その均等物とによって定められ、詳細な説明に記載された事項によって定められるものではない。

【0064】以上の説明に関して更に以下のような態様が考えられる。

【0065】(付記1) RTLデジタル回路の割当決定図を生成する手順と、割当決定図内のモジュールを識別する手順と、9値記号代数を使用して、識別されたモジュールに対する目標を決定する手順と、テスト環境を獲得するため、割当決定図をトラバースすることにより、目標を正当化し伝播させる手順と、獲得されたテスト環境を使用して、所定のテストベクトルをRTL回路に適用する手順と、を有するRTLデジタル回路をテストするテストパターン生成方法。(1)。

【0066】(付記2) 割当決定図を生成する手順は、RTLデジタル回路のハードウェア記述言語表現を受け取る手順と、RTLデジタル回路のハードウェア記述言語表現を割当決定図に変換する手順と、を有する、付記1記載のテストパターン生成方法。

【0067】(付記3) 割当決定図は、RTLデジタル回路の機能的動作を表現するグラフ状構造部と、RTLデジタル回路の近似的な構造表現部と、を有する、付記1又は2記載のテストパターン生成方法。

【0068】(付記4) 識別されたモジュールは、算術演算、論理アレイ、ランダム論理ブロック、記憶素子、内部接続及びブラックボックスを含む、付記1乃至3のうちいずれか一項記載のテストパターン生成方法。

【0069】(付記5) 目標は、RTLデジタル回路並びに識別されたモジュールの入力側の値及び出力側の値と、フレームサイクルとを表現する記号変数を含む、付記1乃至4のうちいずれか一項記載のテストパターン生成方法。

【0070】(付記6) 目標は、RTLデジタル回路並びに識別されたモジュールの入力側の値及び出力側の値と、フレームサイクルと、状態とを表現する記号変数を含む、付記1乃至4のうちいずれか一項記載のテストパターン生成方法。

【0071】(付記7) 9値記号代数は、ある値を、値と状態の一方へ制御する能力を表現する記号変数を含む、付記1乃至6のうちいずれか一項記載のテストパターン生成方法。

【0072】(付記8) 9値記号代数は、値を観測する能力を表現する記号変数を含む、付記7記載のテストパターン生成方法。

【0073】(付記9) 9値記号代数は、1への可制御性、0への可制御性、全て1への可制御性、定数への可制御性、状態への可制御性、任意の値への可制御性、高インピーダンス状態への可制御性、障害の可観測性、及び、障害のコンプリメント可観測性により構成された記号変数を含む、付記1乃至6のうちいずれか一項記載のテストパターン生成方法。

【0074】(付記10) 目標を正当化し伝播させる手順は、RTLデジタル回路の主入力から識別されたモジュールの入力へのテストベクトルを正当化する手順と、識別されたモジュールの出力からRTLデジタル回路の主出力へテスト応答を伝播させる手順と、を有する、付記1乃至9のうちいずれか一項記載のテストパターン生成方法。

【0075】(付記11) 目標を正当化し伝播させる手順は、バックトラック及び時間制限を有するブランチ・アンド・バウンド探索手続を含む、付記1乃至9のうちいずれか一項記載のテストパターン生成方法。

【0076】(付記12) テスト環境は、識別されたモジュールの主入力から主出力までの正当化パス及び伝播パスのセットを含み、テストベクトルが識別されたモジュールの主入力に供給されたとき、識別されたモジュールを動作させる、付記1乃至11のうちいずれか一項記載のテストパターン生成方法。

【0077】(付記13) 正当化パス及び伝播パスは多数のクロックサイクルに亘って進む、付記12記載のテストパターン生成方法。

【0078】(付記14) テスト環境を獲得するため発見的手法を利用する手順を更に有する、付記1記載のテストパターン生成方法。(2)。

【0079】(付記15) モジュールを識別し、目標を決定し、目標を正当化し伝播させ、識別された各モジュールに対しシステムレベル・テストセットを獲得するため、獲得されたテスト環境毎に識別された各モジュールに所定のテストベクトルを適用することを継続する手順を更に有する、付記1記載のテストパターン生成方法。(3)。

【0080】(付記16) RTLデジタル回路用の完全テストセットを獲得するため、識別されたモジュール毎にシステムレベル・テストセットを連結する手順を更に有する、付記15記載のテストパターン生成方法。(4)。

【0081】(付記17) テストベクトルはテストセットライブラリから獲得される、付記1乃至15記載のうちいずれか一項記載のテストパターン生成方法。

【0082】(付記18) 単一クロック線を有するデジタル回路の機能的RTLレベル回路設計情報を取得する手順と、取得した回路設計情報からデータ構造を生成する手順と、生成されたデータ構造から論理構造部及びブロックを識別する手順と、正当化パスを識別するた

め、論理構造部及びブロックの入力からRTLデジタル回路の主入力まで、テスト目標を正当化する手順と、伝播パスを識別するため、論理構造部及びブロックの出力からRTLデジタル回路の主出力までテスト目標を伝播させる手順と、識別された正当化パス及び伝播パスを用いて、所定のテストベクトルをデジタル回路に適用する手順と、を有する、RTLデジタル回路をテストするテストパターン生成方法。(5)。

【0083】(付記19) データ構造を生成する手順は、ハードウェア記述言語表現を用いてRTLデジタル回路設計情報を表現する手順と、ハードウェア記述言語表現を割当決定図に変換する手順と、を有する、付記18記載のテストパターン生成方法。

【0084】(付記20) 割当決定図は、RTLデジタル回路の機能的動作を表現するグラフ状構造部と、RTLデジタル回路の近似的な構造表現部と、を有する、付記19記載のテストパターン生成方法。

【0085】(付記21) 識別された論理構造部は、算術演算、論理アレイ、ランダム論理ブロック、記憶素子、内部接続及びブラックボックスを含む、付記18乃至20のうちいずれか一項記載のテストパターン生成方法。

【0086】(付記22) テスト目標を正当化する手順とテスト目標を伝播させる手順は、9値記号代数を用いて決定された目標を使用する手順を含む、付記18乃至21のうちいずれか一項記載のテストパターン生成方法。(6)。

【0087】(付記23) 目標は、RTLデジタル回路並びに識別された論理構造部の入力側の値及び出力側の値と、フレームサイクルとを表現する記号変数を含む、付記22記載のテストパターン生成方法。

【0088】(付記24) 目標は、RTLデジタル回路並びに識別された論理構造部の入力側の値及び出力側の値と、フレームサイクルと、状態とを表現する記号変数を含む、付記22記載のテストパターン生成方法。

【0089】(付記25) 9値記号代数は、ある値を、値と状態の一方へ制御する能力を表現する記号変数を含む、付記22項記載のテストパターン生成方法。

【0090】(付記26) 9値記号代数は、値を観測する能力を表現する記号変数を含む、付記25記載のテストパターン生成方法。

【0091】(付記27) 9値記号代数は、1への可制御性、0への可制御性、全て1への可制御性、定数への可制御性、状態への可制御性、任意の値への可制御性、高インピーダンス状態への可制御性、障害の可観測性、及び、障害のコンプリメント可観測性からなる9個の記号変数を含む、付記22記載のテストパターン生成方法。

【0092】(付記28) 正当化パス及び伝播パスを識別するため発見的手法を利用する手順を更に有する、

21

付記18記載のテストパターン生成方法。

【0093】(付記29) 論理構造部及びブロックを識別し、テスト目標を正当化し、テスト目標を伝播させ、識別された論理構造部及びブロック毎にシステムレベル・テストセットを獲得するため、識別された正当化パス及び伝播パス毎に識別された論理構造部及びブロックに対し所定のテストベクトルを適用することを継続する手順を更に有する、付記18記載のテストパターン生成方法。(7)。

【0094】(付記30) RTLデジタル回路用の完全テストセットを獲得するため、識別された論理構造部及びブロック毎にシステムレベル・テストセットを連結する手順を更に有する、付記29記載のテストパターン生成方法。(8)。

【0095】(付記31) テストベクトルはテストセットライブラリから獲得される、付記18乃至30のうちいずれか一項記載のテストパターン生成方法。る手順を更に有する、請求項7記載のテストパターン生成方法。

【0096】(付記32) RTLデジタル回路の割当決定図を生成する機能と、割当決定図内のモジュールを識別する機能と、9値記号代数を使用して、識別されたモジュールに対する目標を決定する機能と、テスト環境を獲得するため、割当決定図をトラバースすることにより、目標を正当化し伝播させる機能と、獲得されたテスト環境を使用して、所定のテストベクトルをRTL回路に適用する機能と、をコンピュータに実行させるためのRTLデジタル回路用のテストパターン生成プログラム。(9)。

【0097】(付記33) 単一クロック線を有するデジタル回路の機能的RTLレベル回路設計情報を取得する機能と、取得した回路設計情報からデータ構造を生成する機能と、生成されたデータ構造から論理構造部及びブロックを識別する機能と、正当化パスを識別するため、論理構造部及びブロックの入力からRTLデジタル回路の主入力まで、テスト目標を正当化する機能と、伝播パスを識別するため、論理構造部及びブロックの出力からRTLデジタル回路の主出力までテスト目標を伝播

22

させる機能と、識別された正当化パス及び伝播パスを用いて、所定のテストベクトルをデジタル回路に適用する機能と、をコンピュータに実行させるためのRTLデジタル回路用のテストパターン生成プログラム。(10)。

【0098】

【発明の効果】本発明のテストパターン生成方法は、RTL設計におけるデータパス上の論理値を縮退させた9個のシンボルを用いて値の伝搬処理及び正当化処理を簡便にすることによって大規模なデータパスを含むRTL回路のテスト生成を高速かつ高品質に実現する。

【0099】また、本発明によれば、デジタル回路の論理レベルでスタック・アット障害を適切に取り扱う大規模順序回路用のテストパターンを生成することが可能になり、これにより、設計プロセスの単純化や、設計所要時間の短縮が達成される。

【図面の簡単な説明】

【図1】本発明によるテストパターン生成方法の一実施例のフローチャートである。

【図2】本発明のテストパターン生成方法で使用される割当決定図の一実施例を示す図である。

【図3】ハードウェア記述言語で表現された機能的RTL回路の一実施例を示す図である。

【図4】図3に表現されたRTL回路の割当決定図の一実施例を示す図である。

【図5】図3に表現されRTL回路に対する状態遷移図である。

【図6】割り当て決定回路の内部の正当化及び伝播探索を実行するサブプロセスの一実施例のフローチャートである。

【図7】図6に示されるようなサブプロセスで記憶、操作される目標の一実施例のブロック図である。

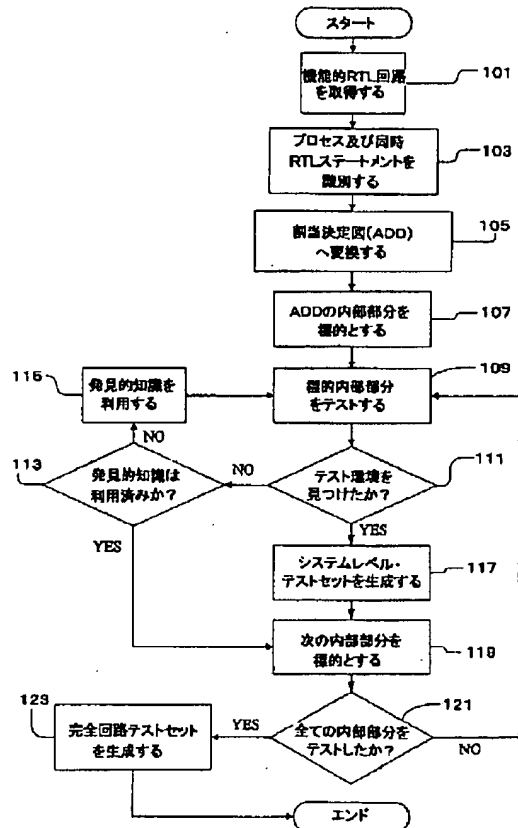
【図8】発見されたテスト環境をテストするサブプロセスの一実施例のフローチャートである。

【図9】デジタル回路内で識別された記憶素子をテストするサブプロセスの一実施例のフローチャートである。

【図10】デジタル回路内で識別されたブラックボックスをテストするサブプロセスのフローチャートである。

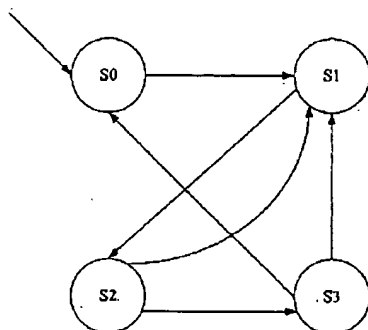
【図1】

本発明によるテストパターン生成方法の一実施例のフローチャート



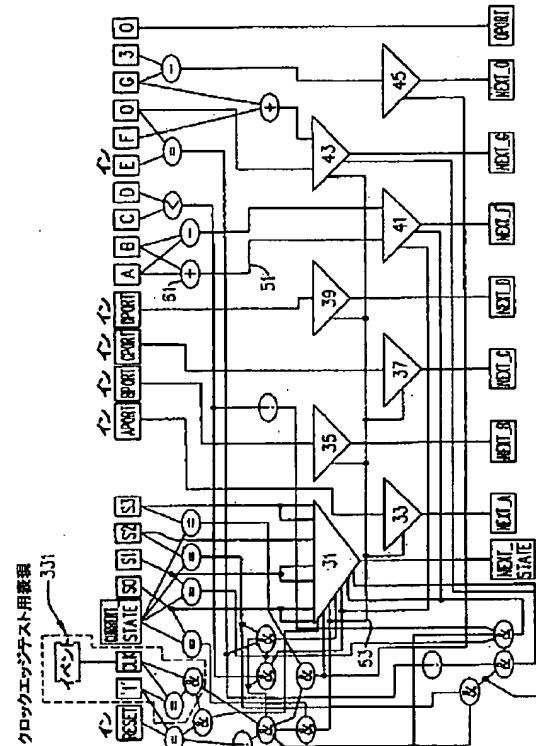
【図5】

図3に表現されたRTL回路に対する状態遷移図



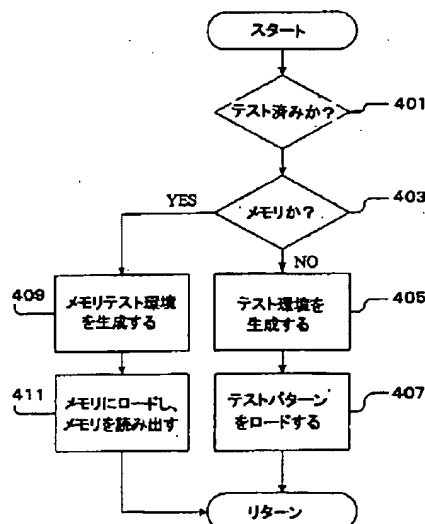
【図4】

図3に表現されたRTL回路の割当決定図の一実施例を示す図



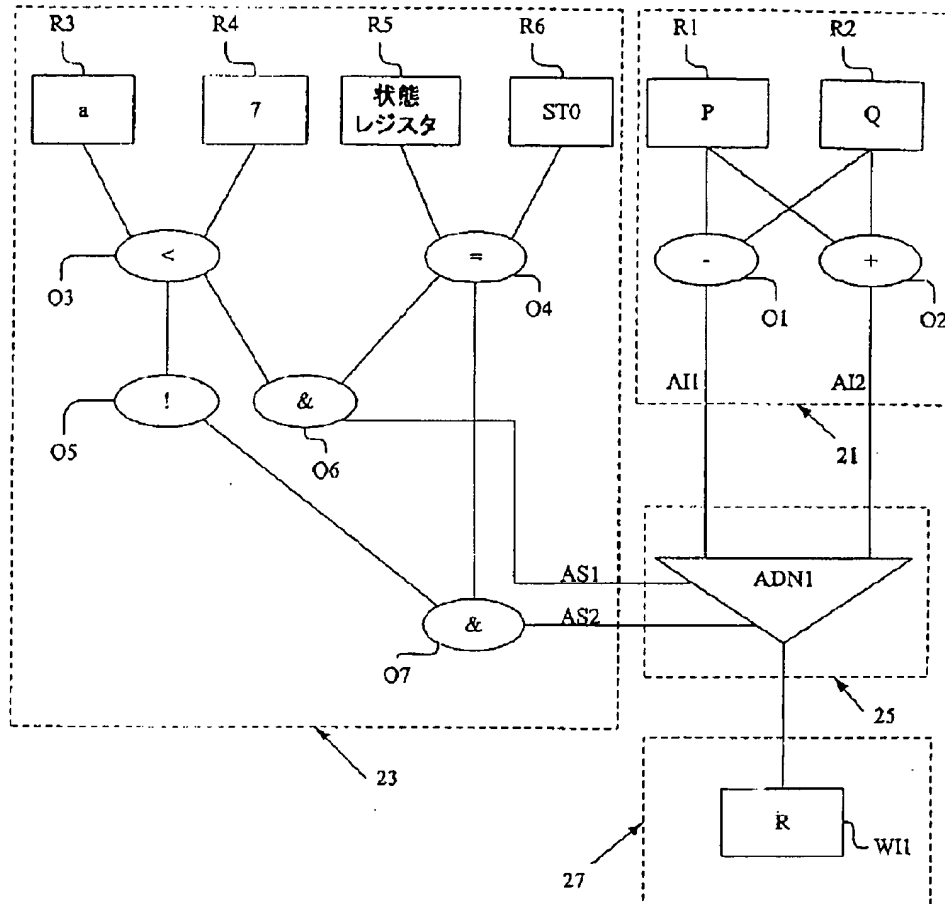
【図9】

記憶素子をテストするサブプロセスの一実施例のフローチャート



【図2】

本発明のテストパターン生成方法で使用する割当決定図の一実施例を示す図



【図3】

ハードウェア記述言語で表現された機能的RTL回路の一実施例を示す図

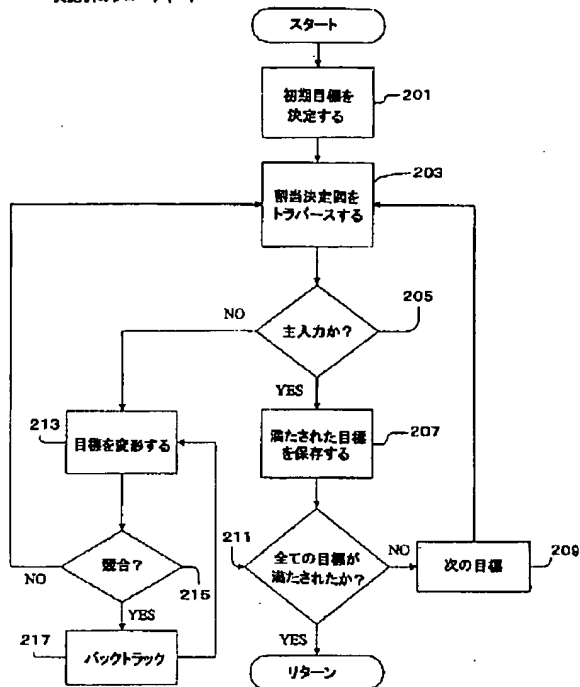
```

entity T1 is
  port ( RESET, CLK : IN std_logic;
        APORT, BPORT, CPORT, DPORT : IN std_logic_vector (7 downto 0);
        E : IN std_logic;
        OPORT : OUT std_logic_vector (7 downto 0));
end T1;
architecture RTL of T1 is
  type STATE_TYPE is ( S0, S1, S2, S3);
  signal CURRENT_STATE, NEXT_STATE : STATE_TYPE;
  signal A, B, C, D, F, G, O, NEXT_A, NEXT_B, NEXT_C, NEXT_D,
        NEXT_F, NEXT_G, NEXT_O : std_logic_vector (7 downto 0);
begin
  COMBIN : process (CURRENT_STATE)
  begin
    NEXT_A <= A; NEXT_B <= B; NEXT_C <= C; NEXT_D <= D;
    NEXT_F <= F; NEXT_G <= G; NEXT_O <= O; OPORT <= O;
    case CURRENT_STATE is
      when S0 =>
        NEXT_A <= APORT; NEXT_B <= BPORT;
        NEXT_C <= CPORT; NEXT_D <= D;
        NEXT_G <= "00000000";
      when S1 =>
        if ( C < D ) then
          NEXT_F <= A + B;
          NEXT_STATE <= S2;
        else
          NEXT_F <= A - B;
          NEXT_STATE <= S3;
        end if;
      when S2 =>
        NEXT_G <= F + G;
        if ( E = '0' ) then
          NEXT_STATE <= S1;
        else
          NEXT_STATE <= S3;
        end if;
      when S3 =>
        NEXT_O <= G - 3;
        NEXT_STATE <= S0;
      end case;
    end process;
  SYNCH : process (CLK, RESET)
  begin
    if ( CLK'event and CLK = '1' ) then
      if ( RESET = '1' ) then
        CURRENT_STATE <= S0;
      else
        A <= NEXT_A; B <= NEXT_B; C <= NEXT_C; D <= NEXT_D;
        G <= NEXT_G; F <= NEXT_F; O <= NEXT_O;
      end if;
    end if;
  end process;
end RTL;

```

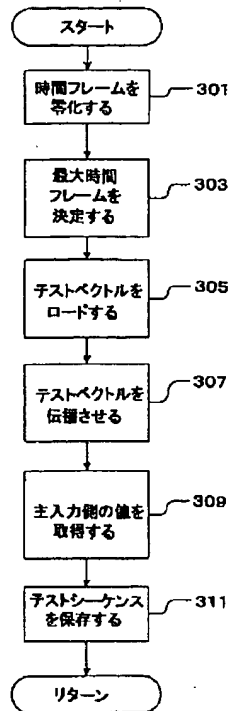
【図6】

割り当て決定回路の内部の正当化及び伝播探検を実行するサブプロセスの一実施例のフローチャート



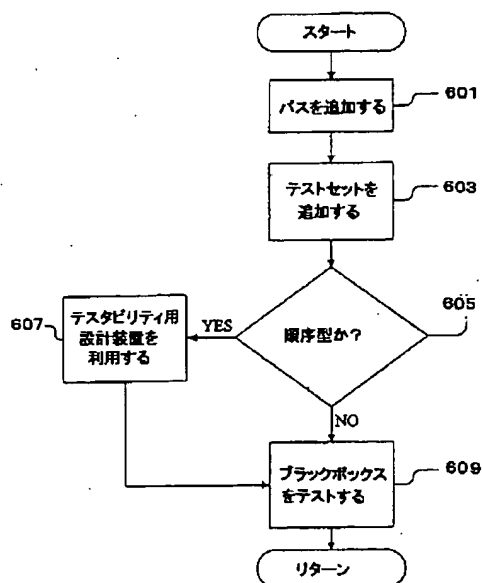
【図8】

テスト環境をテストするサブプロセスの一実施例のフローチャート



【図10】

ブラックボックスをテストするサブプロセスのフローチャート



【図7】

サブプロセスで記憶、操作される目標の一実施例のブロック図

